(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 May 2001 (25.05.2001)

PCT

(10) International Publication Number
WO 01/37496 A1

(51) International Patent Classification[7]: H04L 12/58, 29/06

(21) International Application Number: PCT/IE00/00140

(22) International Filing Date:
16 November 2000 (16.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
990956 16 November 1999 (16.11.1999) IE

(71) Applicant *(for all designated States except US)*: **ORAD SOFTWARE LIMITED** [IE/IE]; Viasec House, Donegal Town, County Donegal (IE).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LAKHDAR, Adly** [DE/IE]; Donegal Town, County Donegal (IE). **MCMULLIN, Maurice** [IE/IE]; Donegal Town, County Donegal (IE).

(74) Agents: **WELDON, Michael, J. et al.**; c/o John A. O'Brien & Associates, Third floor, Duncairn House, 14 Carysfort Avenue, Blackrock, County Dublin (IE).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DE (utility model), DK, DK (utility model), DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *With international search report.*
— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WO 01/37496 A1

(54) Title: A SECURE EMAIL DELIVERY SYSTEM

(57) Abstract: A secure email delivery system (1) acts as a relay between a client computer (2) using Web mail services on a mail server (4) and with the mail server (4). It thus handles incoming and outgoing messages. A policy manager (74) determines a security policy from a policy database (30) when an outgoing message is received. According to this policy, the message may be signed and encrypted transparently to the user. All registered users have a policy and the system captures security data for communication with non-registered users when they reply to outgoing messages from registered users.

"A secure email delivery system"

INTRODUCTION

Field of the Invention

The invention relates to delivery of secure email messages.

Prior Art Discussion

It is well known that a major barrier to use of email for delivery of business and legal documents is the problem of ensuring security of messages and attached documents. This problem remains despite the fact that major advances in cryptography have been made in recent years. It appears that many people have not used the available key pair and digital signature systems because there is a perception that they require excessive processor time and are difficult to use.

The invention addresses this problem.

SUMMARY OF THE INVENTION

According to the invention, there is provided a secure email delivery system comprising:

a client-side interface comprising means for receiving outgoing messages from client devices operated by registered subscribers and for transmitting incoming messages to said client devices;

a policy database storing a security policy for each of a plurality of registered subscribers;

a policy manager comprising means for accessing the policy database to determine a registered subscriber policy in real time and for delivering policy data to a requesting function in real time;

a secure key database securely storing keys of registered subscribers and their addressees and a secure certificate database securely storing certificates of registered subscribers and their addressees;

a signing function comprising means for receiving policy data from the policy manager and a registered subscriber private key from the key database and for dynamically signing an outgoing message received at the client-side interface;

an encryption function comprising means for receiving policy data from the policy manager and for dynamically encrypting an outgoing message in real time using an addressee public key retrieved from the certificate database;

a decryption function comprising means for automatically decrypting an incoming message in real time using the private key of the addressee registered subscriber retrieved from the key database; and

a server-side interface comprising means for transmitting outgoing messages to a mail server after being processed, and for receiving incoming messages.

In one embodiment, the policy database stores a policy for non-registered users who are addressees of mail from registered users, and the policy manager comprises means for determining policy data according to both a sender and a recipient of a message.

In one embodiment, the signing function comprises means for automatically signing an outgoing message in the absence of policy data for an addressee.

In one embodiment, the client-side interface comprises an API for linking with hooks on a server.

In one embodiment, the API comprises means for linking with hooks on an external mail server for generating Web mail under client instructions.

In one embodiment, the policy manager is linked with the signing function, the encryption function, and with the decryption function by a cryptography API providing a real time bi-directional link and allowing modularity of the functions of each side of said API.

In another embodiment, the signing function, the encryption function, and the decryption function comprise means for operating with use of a cryptography library of low-level cryptography processes.

In one embodiment, the system further comprises a certificate manager and a key manager residing on the same side of the cryptography API as the signing, encryption and decryption functions.

In another embodiment, the signing function, the encryption function, the decryption function, the certificate manager, and the key manager provide a programmed wrapper around the cryptography library.

In one embodiment, the key database stores a policy as an instance of a plurality of datatype classes.

In one embodiment, the datatype classes include signing algorithm, encryption algorithm, signing policy, and decryption policy classes.

In one embodiment, the key database stores a key pool in a directory having a configurable size with a threshold, and the database comprises means for replenishing keys when the level falls below the threshold.

In another embodiment, a management console comprises means for editing data in the policy database.

In one embodiment, the management console comprises means for retrieving a policy from the policy database as a model and for retrieving a certificate from the key database as a model.

In one embodiment, the management console comprises a controller comprising means for treating each business operation as an object-orientated class instance.

In another embodiment, the controller comprises means for receiving a command string, parsing the string into a hash table, instantiating a class instance and setting properties and a name for the instance, and initialising the instance.

In one embodiment, the policy database stores group policies, each associated with a group of registered subscribers.

In another embodiment, the policy database stores default policies.

In one embodiment, the certificate manager comprises means for automatically stripping certificates from incoming messages and storing them in the certificate database for subsequent use.

According to another aspect, the invention provides a secure email delivery system comprising:

a client-side interface comprising means for receiving outgoing messages from client devices operated by registered subscribers and for transmitting incoming messages to said client devices;

a policy database storing a security policy for each of a plurality of registered subscribers;

a policy manager comprising means for accessing the policy database to determine a subscriber policy in real time and for delivering policy data to a requesting function in real time;

a secure key database securely storing keys of registered subscribers and their addressees and a secure certificate database securely storing certificates of registered subscribers and their addressees;

a signing function comprising means for receiving policy data from the policy manager and a subscriber private key from the key database and for dynamically signing an outgoing message received at the client-side interface;

an encryption function comprising means for receiving policy data from the policy manager and for dynamically encrypting an outgoing message in real time using an addressee public key retrieved from the certificate database;

a decryption function comprising means for automatically decrypting an incoming message in real time using the private key of the addressee retrieved from the key database; and

a server-side interface comprising means for transmitting outgoing messages to a mail server after being processed, and for receiving incoming messages; and wherein:

the policy manager is linked with the signing function, the encryption function, and with the decryption function by a cryptography API providing a real time bi-directional link and allowing modularity of the functions on each side of said API,

the signing function, the encryption function, and the decryption function comprises means for operating with use of a cryptography library of low-level cryptography processes, and

the system further comprises a key manager and a certificate manager residing on the same side of the API as said signing, encryption, and decryption functions, and said functions, the key manger, and the certificate manager together provide a wrapper around said cryptography library.

DETAILED DESCRIPTION OF THE INVENTION

Brief Description of the Drawings

The invention will be more clearly understood from the following description of some embodiments thereof given by way of example only with reference to the accompanying drawings, in which:

Fig. 1 is a high level diagram showing interaction of a security relay of the invention with a client computer and with a mail server;

Fig. 2 is a diagram showing interaction between the client computer and the security relay in more detail;

Fig. 3 is a diagram showing the security relay in more detail;

Figs. 4 and 5 are diagrams showing parts of the relay in more detail; and

Fig. 6 is a flow diagram illustrating operation of the relay in more detail.

Description of the Embodiments

Referring to Fig. 1, a secure email delivery system 1 interfaces with a client computer 2 over a secure HTTP socket layer (SSL) connection 3. It also interfaces with a local mail server 4 via APIs using a HTTP SSL link 5. The mail server 4 sends and receives messages over the Internet 6. Because the system 1 effectively operates as a relay between the client and server, it is henceforth called a" security relay" or "relay".

The client computer 2 is conventional insofar as it only needs to have a messaging application or browser which supports a portable code environment such as a Java Virtual Machine™ environment. The computer 2 is operated by a registered subscriber, also referred to as a "user". The mail server is a conventional mail server for an ISP. In this embodiment, the relay 1 resides on the same platform as the mail server 4.

Referring to Fig. 2, the client computer 2 interfaces with a Web server 10 hosting the mail server 4 and the relay 1. The client computer 2 runs a browser 11 having a portable code interpreter which allows execution of plug-ins 12. The Web server 10 allows the client computer 2 to communicate with the mail server 4 to create "Web mail" messages while on-line. The server 4 then sends the messages to the recipients

and receives the incoming messages in a mailbox for the user. The user then subsequently retrieves the messages. The security relay transparently to the user performs a set of desired security operations for both incoming and outgoing messages and so it may be regarded as conceptually residing between the client computer 2 and the mail server 4. However, the physical interconnection is via APIs between the mail server 4 and the relay 1, this channel transmitting:

(a) outgoing messages from the mail server 4 to the relay 1 and security-processed outgoing messages back to the mail server 4 for onward transmission to the recipient, and

(b) incoming messages from the mail server 4 and security-processed incoming messages back to the mail server 4 for onward transmission to the client computer 2.

A key feature is that the relay 1 handles all cryptography and digital signature operations for the client 2. These operations are carried out transparently to the user and require no input from him or her. Thus a user only needs to subscribe to the security service provided by an organisation hosting the relay 1. Such an organisation may be any ISP or ASP, as the Internet is of course the major insecure network in use. Thus the (real or perceived) problem of setting up encryption/decryption functionality is taken from the user. He or she only needs to send and receive email messages in the usual manner. The following is a typical message sequence:

User X (of computer 2) sends a message to a remote user Y. This is created over the secure socket 3 by the client computer 2 and the mail server 4.

The message is routed to Y via the relay 1. This is not encrypted, but is signed automatically by the relay 1 on behalf of X.

If Y responds, the relay 1 captures Y's certificate on the return path. From then on the relay 1 will automatically encrypt and sign all outgoing messages to Y, and also decrypt all incoming messages.

Thus, the user X has a secure bi-directional link with Y without the need to bother with any cryptography operations.

Because the relay1 interfaces with an existing mail server 4, it is particularly convenient for the ISP or ASP hosting it. The relay 1 can be added in a modular manner.

Referring now to Fig. 3 the relay 1 is shown in more detail. It interfaces on the server side with (a) an SMTP delivery module 20 and an SMTP server 21 for sending messages, and (b) with an IMAP/POP3 mail server 22 and an IMAP/POP3 retriever 23 for receiving messages. This interfacing is via APIs executing on the platform which hosts the relay 1 and the mail server 4 as applications.

A security policy database 30 allows the relay 1 to provide the security required by users according to policies set by a system administrator. This database is managed by a management console function 31.

A user authentication module 35 uses the database 30 to dynamically perform authentication of subscribed users. A client side interface API function block 41 receives messages via the client plug-ins 12.

The messages are passed to encryption functions 37 which perform encryption and digital signing according to the policies for the users and addressees retrieved from the database 30. As described above, if the addressee is being addressed for the first time there is digital signing by default, however the outgoing message can not be

encrypted. However, once a reply is received from the addressee his or her certificate is stored in the database 30 and there will be encryption/decryption from then on.

Messages are retrieved from the retriever 23 and are decrypted where applicable and the signature is authenticated by functions 38 using the database 30. The header (sender, subject, date) is passed to the client computer 2 for display (after decryption) if it is selected by the user.

In more detail, referring to Fig. 4 the policy database 30 and the management console 31 are illustrated. They may be together referred to as an administrative framework. The framework follows the MVC (Model-View-Controller) design pattern, allowing the management presentation interface to be completely decoupled from the logic. Each retrieved policy is represented as a JavaBean model 53 and each retrieved certificate is represented as a model 54. A certificate interface 56 is linked with a certificate/key database 90, shown in Fig. 5. A view interface 50 uses HTML with embedded JSP tags 51. Post/get control is implemented by a set of Java servlets 52.

Referring to Fig. 5, the functions 37 and 38 are illustrated in detail. A Java cryptography API 75 is linked with a policy manager 74 and it has a HTTP client 76 linked with:

a fast Common Gateway Interface (CGI) cryptography server 77 programmed in C++,
a S/MIME C++ cryptography function 78,
a certificate manager 79 programmed in C++,
a key manager 80 programmed in C++, and
a cryptography library 81 having access to a certificate/key database 90.

The interface 41 comprises:

an API 71,
a HTTP client 72 programmed in native Java, and
an API servlet 73, and

In more detail, the management console 31 provides graphical user interfaces for input and display of user information, security policies, and certification management. The view interface 50 is the console front end and the screens are displayed using HTML and JSP. The JSP tags 51 are used to interface between the screens and Java ode for the purposes of displaying derived data that is calculated or changes during the lifetime of the relay 1. The controller servlets provide the derived data to the JSP tags. The security policy model 53 is a Javabean which drives the data for the security policy screens. Likewise, the certificate model 54 is a Java bean which drives data for certificate management screens. The function 55 provides access to the database 30 for persistent data on policies using the Java standard JDBC. The certificate interface interacts with the key database 90 to retrieve certificate data for the management console 31. Access to the database 90 is via the cryptography library 81, described in more detail below.

Referring again to Fig. 5, the functions which perform the core cryptography operations are now described in detail. The mail server 4 shown in Fig. 1 includes a client 70 which accesses the relay 1 via the client interface 10. In this example, the client of the server 4 is written in Perl/PHP. The client of the server calls the API 71 of the block 41 and the HTTP client 72 provides a mechanism for the client hooks 70 to communicate with the relay 1. This mechanism involves use of the API servlet 73 for server-side processing.

The policy manager 74 determines what security policies should be applied to an email being received or sent. It does this by communicating with the function 55, which extracts information from the database 30.

The cryptography API 75 is an interface which provides a real time bi-directional link and allows Java code to call C++ cryptography code. The cryptography API client 76 allows the cryptography library 81 to be distributed on a different hardware platform for fault tolerance, versatility, and performance. The Fast CGI server 77 is the server side of the cryptography API 75.

The component 78 provides S/MIME cryptography functionality for signing, encryption, and decryption. The component 79 provides certificate management functionality, and the component 80 provides key pool management functionality. These components interact with the key database 90, and use cryptography processes of the library 81. The library 81 implements low-level cryptography algorithms and key generation routines. The components 78, 79, and 80 provide a C++ wrapper around the low-level C library 81.

In the architecture illustrated in Fig. 4, the controller 52 manages the flow of control for its operations by receiving a CGI command for instantiating business logic, and each business logic instance is a Java bean class instance. The control flow is as follows:

Taking the command string from the CGI variables,
Parsing the string into a hash table,
Instantiating a bean using introspection and the command string,
Setting the bean properties from the name, value pairs in the hash table,
Calling *init()* on the bean, and
Pushing the data into the view for display.

The policy database 30 stores policies in which a policy consists of one instance of each of the following data type classes:

Security policy scope,

Signing algorithm,
Encryption algorithm,
Signing policy,
Encryption policy

The relay 1 automatically captures security data for non-registered users when they reply to an outgoing message from a registered user. This is achieved by the certificate cryptography function 78 stripping off the certificate from a signed message and storing the certificates in the database 90.

The key pool database 90 is structured to have minimal impact on performance of the relay 1 as generation of key pairs is processor-intensive. The key pool directory size is configurable by the user, and a threshold is set for the directory and when the number of keys falls below that level they are replenished. The key pool is replenished with anonymous key pairs, but these are still fully usable keys sets. The key sets are made usable by the generation of certificates and this happens on demand. A daemon is used to monitor key pool threshold levels, and message queues are used to communicate between the controller and the key pool code.

Referring now to Fig. 6 operation of the relay 1 is described. The client computer 2 generates an email using Web server functions 10 and this is processed by the module 37. The plug-ins 12 allow the client computer 2 to route the email to the relay 1 via the mail server Web mail functionality. This is a very effective mechanism for integration of the relay 1 with a mail server 4 in a modular manner. With an open API any subscriber or administrative user can interface with the relay 1 by implementing a plug-in 12 so that its code interacts with the API 41.

In the module 37, the security policy is determined based on the "from" and "to" addresses in the message. The security policy consists of:

Digital Signature Policy

Do not sign

Clear-sign

Opaque sign

Digest Algorithm

Encryption policy

Do not encrypt

Encrypt

(Symmetric) encryption algorithm.

The policy manager 74 accesses the policy database 30 to get policies for the senders and receivers of messages. Each policy contains a "from" and a "to" field. Some policies contain wildcards to allow setting of policies for groups. The relay 1 has default policies, and these can be changed by the system administrator. The policies contain the signing and encryption polices and algorithms.

The key manager 80 then obtains access to the sender's private key from the database 90. This is secure because the keys are stored in the pkc #15 secure storage format.

The S/MIME component 78 then constructs an S/MIME signed message including the digital certificate chain associated with the signing key in the message. Encryption is of course by-passed if the policy does not indicate a requirement for encryption.

If the policy requires encryption, the certificate manager 79 retrieves the recipient certificate from the certificate database. The S/MIME component 78 then constructs an S/MIME encrypted message. The message is encrypted using a randomly-

generated symmetric session key, and the session key is encrypted using the recipient's public key. These cryptography operations are performed by an algorithm selected from the library 81. The structure of the library is as follows:-

key and certificate generation
key and certificate management, and
S/MIME capabilities.

The S/MIME message is then transmitted.

An incoming message is received by the functions 71, 72, 75, 76, 77, and 78. There is no involvement of policies for incoming messages and the messages are decrypted by the S/MIME component 78 and the library 81.

It will be appreciated that the invention provides for very effective security processing of messages in a manner which is modular on the Web server hosted by the ISP or ASP and is transparent to the subscriber. Also, the policy database allows excellent versatility in choice of options by subscribers. The structure of the functions allows excellent versatility for deployment of resources such as the low-level cryptography algorithms. Also, the administrative framework comprising the management console 31 and the policy database 31 allows simple and effective real time configuration by administrative personnel of the host organisation.

The invention is not limited to the embodiments described but may be varied in construction and detail.

Claims

1. A secure email delivery system (1) comprising:

a client-side interface (41) comprising means for receiving outgoing messages from client devices operated by registered subscriber and for transmitting incoming messages to said client devices;

a policy database (30) storing a security policy for each of a plurality of registered subscribers;

a policy manager (74) comprising means for accessing the policy database (30) to determine a registered subscriber policy in real time and for delivering policy data to a requesting function in real time;

a secure key database (90) securely storing keys of registered subscribers and their addressees and a secure certificate database (90) securely storing certificates of registered subscribers and their addressees;

a signing function (78) comprising means for receiving policy data from the policy manager (74) and a registered subscriber private key from the key database (90) and for dynamically signing an outgoing message received at the client-side interface;

an encryption function (78) comprising means for receiving policy data from the policy manager (74) and for dynamically encrypting an outgoing message in real time using an addressee public key retrieved from the certificate database;

a decryption function (78) comprising means for automatically decrypting an incoming message in real time using the private key of the addressee registered subscriber retrieved from the key database (90); and

a server-side interface (71-73) comprising means for transmitting outgoing messages to a mail server after being processed, and for receiving incoming messages.

2. A system as claimed in claim 1, wherein the policy database (30) stores a policy for non-registered users who are addressees of mail from registered users, and the policy manager (74) comprises means for determining policy data according to both a sender and a recipient of a message.

3. A system as claimed in claim 2, wherein the signing function (78) comprises means for automatically signing an outgoing message in the absence of policy data for an addressee.

4. A system as claimed in any preceding claim, wherein the client-side interface comprises an API (71-73) for linking with hooks on a server.

5. A system as claimed in claim 4, wherein the API (71-73) comprises means for linking with hooks on an external mail server for generating Web mail under client instructions.

6. A system as claimed in any preceding claim, wherein the policy manager (74) is linked with the signing function (78), the encryption function (78), and with the decryption function (78) by a cryptography API providing a real time bi-directional link and allowing modularity of the functions of each side of said API.

7. A system as claimed in any preceding claim, wherein the signing function (78), the encryption function (78), and the decryption function (78) comprise means for operating with use of a cryptography library (81) of low-level cryptography processes.

8. A system as claimed in claim 6 or 7, wherein the system further comprises a certificate manager and a key manager residing on the same side of the cryptography API as the signing, encryption and decryption functions.

9. A system as claimed in claim 8, wherein the signing function, the encryption function, the decryption function, the certificate manager, and the key manager provide a programmed wrapper around the cryptography library.

10. A system as claimed in any preceding claim, wherein the key database (90) stores a policy as an instance of a plurality of datatype classes.

11. A system as claimed in claim 10, wherein the datatype classes include signing algorithm, encryption algorithm, signing policy, and decryption policy classes.

12. A system as claimed in claim 11, wherein the key database (30) stores a key pool in a directory having a configurable size with a threshold, and the database comprises means for replenishing keys when the level falls below the threshold.

13. A system as claimed in any preceding claim, further comprising a management console (31) comprising means for editing data in the policy database.

14. A system as claimed in claim 13, wherein the management console (31) comprises means for retrieving a policy from the policy database as a model and for retrieving a certificate from the key database as a model.

15. A system as claimed in claim 14, wherein the management console (31) comprises a controller (52) comprising means for treating each business operation as an object-orientated class instance.

16. A system as claimed in claim 15, wherein the controller (52) comprises means for receiving a command string, parsing the string into a hash table, instantiating a class instance and setting properties and a name for the instance, and initialising the instance.

17. A system as claimed in any preceding claim, wherein the policy database (30) stores group policies, each associated with a group of registered subscribers.

18. A system as claimed in any preceding claim, wherein the policy database stores default policies.

19. A system as claimed in any of claims 8 to 18, wherein the certificate manager comprises means for automatically stripping certificates from incoming messages and storing them in the certificate database for subsequent use.

20. A secure email delivery system (1) comprising:

a client-side interface (41) comprising means for receiving outgoing messages from client devices operated by registered subscribers and for transmitting incoming messages to said client devices;

a policy database (30) storing a security policy for each of a plurality of registered subscribers;

a policy manager (74) comprising means for accessing the policy database (30) to determine a subscriber policy in real time and for delivering policy data to a requesting function in real time;

a secure key database (90) securely storing keys of registered subscribers and their addressees and a secure certificate database (90) securely storing certificates of registered subscribers and their addressees;

a signing function (78) comprising means for receiving policy data from the policy manager (74) and a subscriber private key from the key database (90) and for dynamically signing an outgoing message received at the client-side interface;

an encryption function (78) comprising means for receiving policy data from the policy manager (74) and for dynamically encrypting an outgoing message in real time using an addressee public key retrieved from the certificate database;

a decryption function (78) comprising means for automatically decrypting an incoming message in real time using the private key of the addressee retrieved from the key database (90); and

a server-side interface (71-73) comprising means for transmitting outgoing messages to a mail server after being processed, and for receiving incoming messages; and wherein;

the policy manager (74) is linked with the signing function (78), the encryption function (78), and with the decryption function (78) by a cryptography API providing a real time bi-directional link and allowing modularity of the functions on each side of said API,

the signing function (78), the encryption function (78), and the decryption function (78) comprises means for operating with use of a cryptography library (81) of low-level cryptography processes, and

the system further comprises a key manager (80) and a certificate manager (79) residing on the same side of the API as said signing, encryption, and decryption functions, and said functions, the key manger, and the certificate manager together provide a wrapper around said cryptography library.

21. A computer program product comprising software code portions for completing a secure email delivery system as claimed in claim 1 when executing on a digital computer.

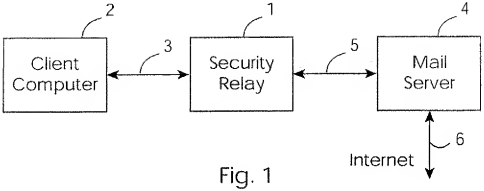
2
1
4
3
5
Client Computer
Security Relay
Mail Server
6
Internet

Fig. 1

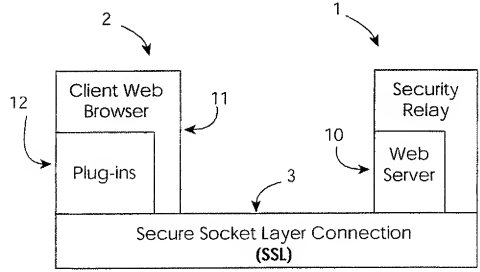
2
1
12
Client Web Browser
11
Security Relay
10
Plug-ins
3
Web Server
Secure Socket Layer Connection
(SSL)

Fig. 2

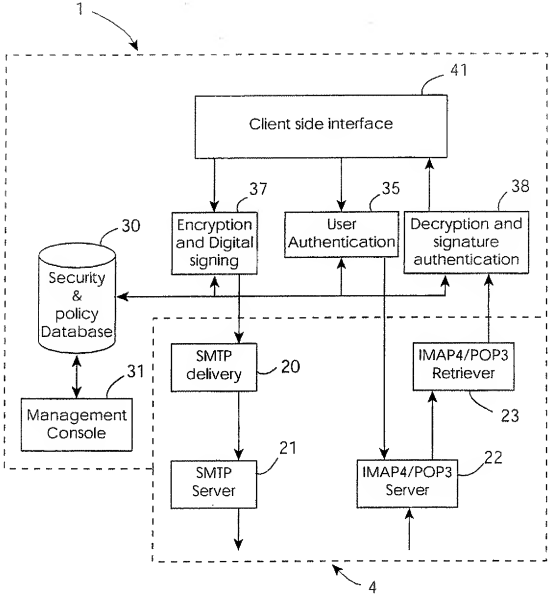
1
41
Client side interface
37
35
38
30
Encryption and Digital signing
User Authentication
Decryption and signature authentication
Security & policy Database
SMTP delivery
20
IMAP4/POP3 Retriever
31
Management Console
23
21
22
SMTP Server
IMAP4/POP3 Server
4

Fig. 3

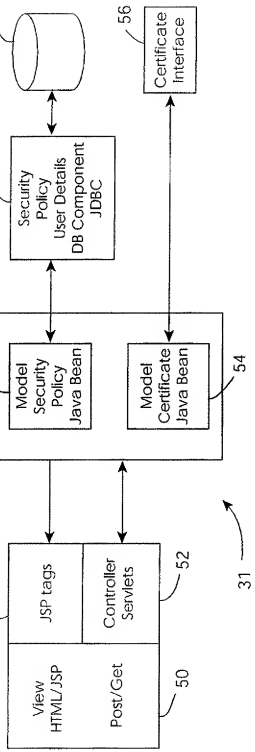
30
56
Certificate Interface
55
Security Policy User Details DB Component JDBC
53
Model Security Policy Java Bean
Model Certificate Java Bean
54
51
JSP tags
Controller Servlets
52
View HTML/JSP
Post/Get
50
31

Fig. 4

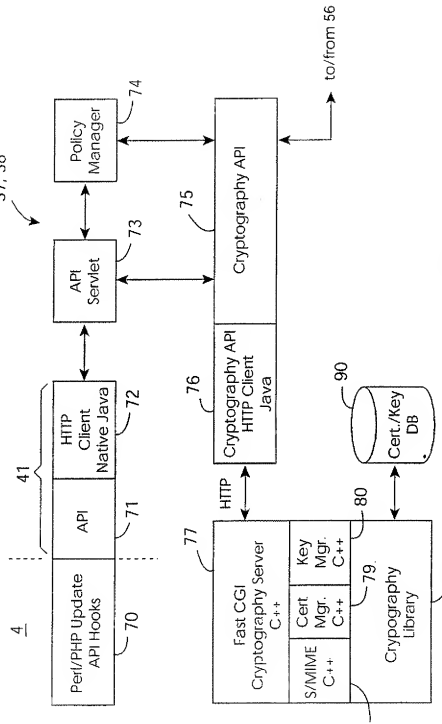
4
Perl/PHP Update API Hooks
70
API
71
41
HTTP Client Native Java
72
37, 38
API Servlet
73
Policy Manager
74
75
Cryptography API
to/from 56
76
Cryptography API HTTP Client Java
HTTP
77
Fast CGI Cryptography Server C++
S/MIME C++
Cert. Mgr. C++
Key Mgr. C++
78
79
80
Cryptography Library
81
90
Cert./Key DB

Fig. 5

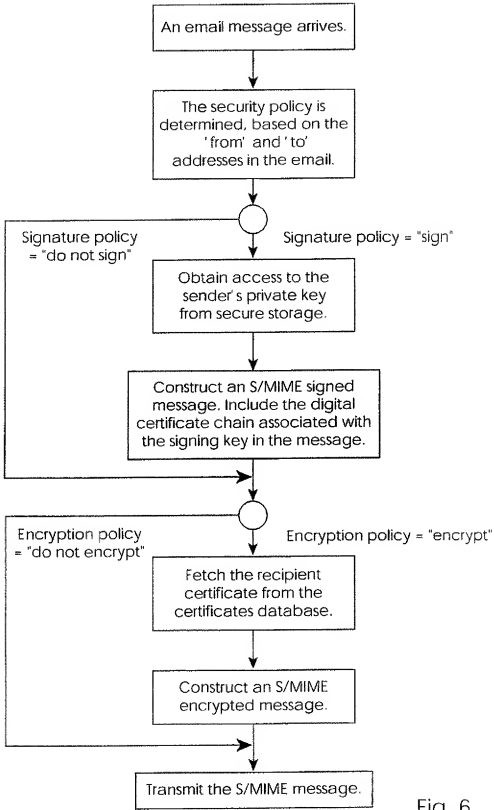
An email message arrives.
The security policy is determined, based on the 'from' and 'to' addresses in the email.
Signature policy = "do not sign"
Signature policy = "sign"
Obtain access to the sender's private key from secure storage.
Construct an S/MIME signed message. Include the digital certificate chain associated with the signing key in the message.
Encryption policy = "do not encrypt"
Encryption policy = "encrypt"
Fetch the recipient certificate from the certificates database.
Construct an S/MIME encrypted message.
Transmit the S/MIME message.

Fig. 6

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IE 00/00140

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L12/58 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 982 856 A (DIMITROFF MICHAEL P ET AL) 9 November 1999 (1999-11-09) abstract figures 1,11,14 column 5, line 35 -column 5, last line column 24, line 34 -column 25, line 36 column 31, line 44 -column 32, last line	1-21
A	KENT S: "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management (URL)" NETWORK WORKING GROUP, 1 February 1993 (1993-02-01), XP002058728 abstract paragraph '0001! paragraph '03.3! paragraph '3.4.2! paragraph '3.4.3!	1-21

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

* Special categories of cited documents :

A document defining the general state of the art which is not considered to be of particular relevance

E earlier document but published on or after the international filing date

L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

O document referring to an oral disclosure, use, exhibition or other means

P document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

& document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 April 2001	17/04/2001

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL – 2280 HV Rijswijk
Tel. (+31–70) 340–2040, Tx. 31 651 epo nl,
Fax: (+31–70) 340–3016

Authorized officer
Cichra, M

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IE 00/00140

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	AUYONG K ET AL: "AUTHENTICATION SERVICES FOR COMPUTER NETWORKS AND ELECTRONIC MESSAGING SYSTEMS" OPERATING SYSTEMS REVIEW (SIGOPS),ACM HEADQUARTER. NEW YORK,US, vol. 31, no. 3, 1 July 1997 (1997-07-01), pages 3-15, XP000739146 paragraph '0003! paragraph '0005! ---	1-21
A	US 5 473 143 A (VAK HUGO ET AL) 5 December 1995 (1995-12-05) abstract column 1, line 63 -column 3, line 9 column 12, line 27 -column 13, line 3 column 14, line 57 -column 16, line 22 column 32, line 34 -column 33, line 45 ---	1,2,4,6, 7,10,13, 17,18, 20,21
A	SUCHUN WU: "MHS SECURITY-A CONCISE SURVEY" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 25, no. 4 / 05, 1 November 1992 (1992-11-01), pages 490-495, XP000306615 ISSN: 0169-7552 the whole document ------	1,2,4,6, 7,10,13, 17,18, 20,21

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/IE 00/00140

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5982856 A	09-11-1999	US 5740231 A	14-04-1998
		AU 3633795 A	09-04-1996
		CA 2199243 A	28-03-1996
		EP 0781482 A	02-07-1997
		US 5872779 A	16-02-1999
		US 6031895 A	29-02-2000
		US 6064723 A	16-05-2000
		WO 9609710 A	28-03-1996
		US 5621727 A	15-04-1997
		US 5761201 A	02-06-1998
US 5473143 A	05-12-1995	US 5265033 A	23-11-1993
		AU 7965094 A	01-05-1995
		CA 2172841 A	13-04-1995
		WO 9510081 A	13-04-1995
		AT 194433 T	15-07-2000
		AU 658590 B	27-04-1995
		AU 9042891 A	27-04-1993
		BR 9107319 A	30-05-1995
		CA 2119563 A,C	01-04-1993
		DE 69132294 D	10-08-2000
		DE 69132294 T	22-02-2001
		EP 0605418 A	13-07-1994
		ES 2151883 T	16-01-2001
		JP 7508362 T	14-09-1995
		WO 9306546 A	01-04-1993